

# Rapport Projet Robot 1

Sophie Oms et Léo-Paul Hauet



## SOMMAIRE

I) Introduction.....	2
II) Alimentation et surveillance alimentation.....	3
III) Capteurs IR.....	6
IV) Affichage LEDs.....	7
V) Moteurs/ Pilotage Rotation Roues-Chenilles.....	8
VI) Affichage bus I2C.....	10
VII) Réception télécommande.....	11
VIII) Sonar et motorisation du sonar par le servomoteur.....	14
IX) Étude de la liaison RS232.....	15
X) Conclusion.....	17

## I) Introduction

Le projet robot a pour objectif d'assembler et de tester un robot. Ce rapport a pour sujet la première partie du projet robot, qui concerne l'assemblage de la carte électronique.

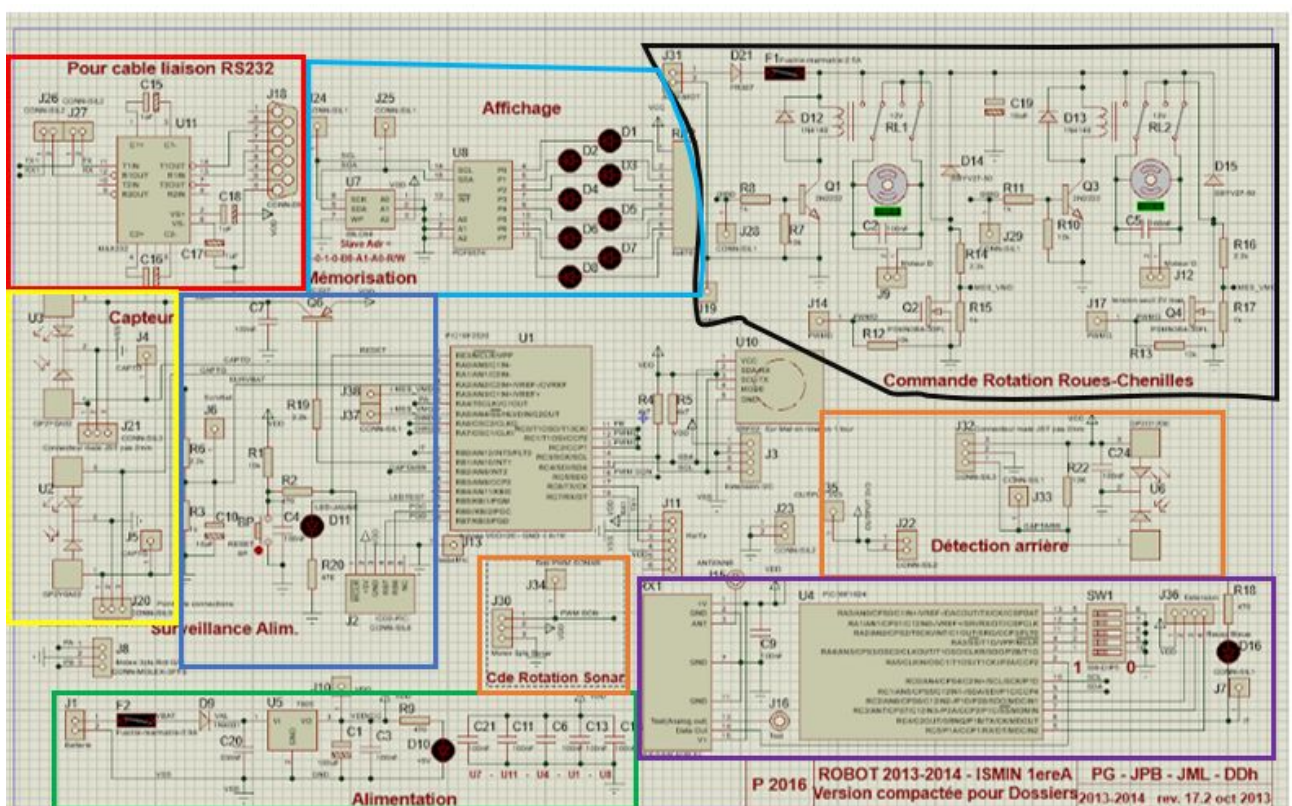
Nous aborderons le fonctionnement de chaque bloc de la carte, en expliquant son rôle. Notre rapport se décompose en dix sous-parties concernant les éléments du robot. Ce robot doit être capable de se déplacer tout en pouvant détecter et éviter des obstacles.

Nous n'avons pas pu achever la réalisation de la carte compte tenu du contexte sanitaire. Par conséquent, nous nous sommes appuyés sur des simulations (logiciel Proteus) pour décrire la logique et le fonctionnement de la carte.

Pour chaque étape, nous avons:

- ❖ repéré et identifié des composants sur le plan de la carte
- ❖ étudié les datasheets en annexe
- ❖ simulé les fonctions sur Proteus pour comprendre leur fonctionnement
- ❖ vérifié le bon fonctionnement des blocs (concordance du circuit réel avec la simulation) pour les deux premiers blocs, qui ont pu être soudés,

### Architecture de la carte du robot :



*Schéma Proteus de la carte robot et de ses différents blocs fonctionnels*

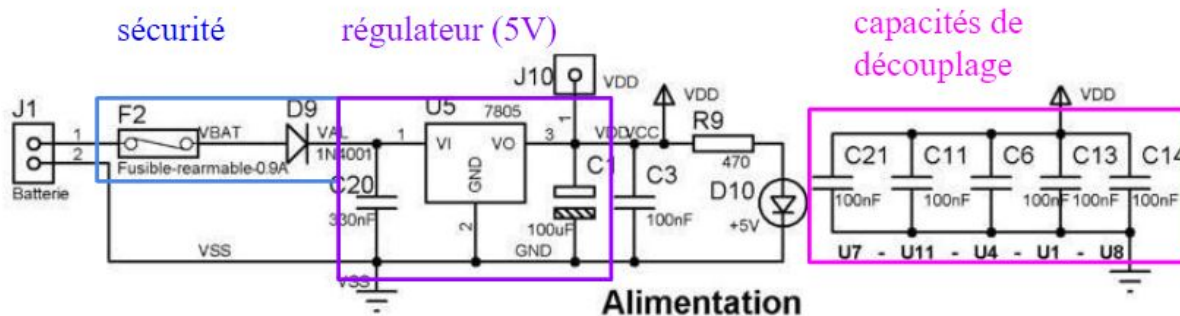
La carte est constituée de 8 blocs, un sur l'alimentation en vert avec sa surveillance et le bouton

reset en bleu foncé, un autre sur les capteurs IR en jaune, un sur l'affichage LED en bleu clair, un sur les moteurs en noir, un sur le sonar et son servo moteur en orange, un sur la télécommande en violet, enfin, un bloc pour les liaisons RS232 en rouge.

## II) Blocs alimentation et surveillance alimentation

### **Bloc alimentation**

Rôle du bloc : Fournir à la carte électronique l'énergie nécessaire à son bon fonctionnement.



Fonctionnement du bloc alimentation: le bloc maintient constamment la tension VDD à 5V par rapport à la masse (VSS), à partir d'une tension continue VBAT venant d'une batterie de 12 Volts placée en J1.

Le bloc alimentation est divisé en plusieurs parties :

La diode D9 et le fusible F2 servent respectivement à protéger la carte d'une inversion de polarité et d'un courant trop élevé qui endommagerait la carte.

Le transistor U5 régule la tension VDD à 5V à l'aide des capacités C20 et C1.

D10 sert de diode témoin pour informer l'utilisateur si le robot est en tension ou pas.

La résistance limite le courant dans le LED D10 pour éviter de la griller.

Enfin, les condensateurs C21, C11, C6, C14, C13 filtrent les signaux carrés pour les lisser et éviter les parasites qui pourraient perturber la carte.

### Vérification de bon fonctionnement de la réalisation :

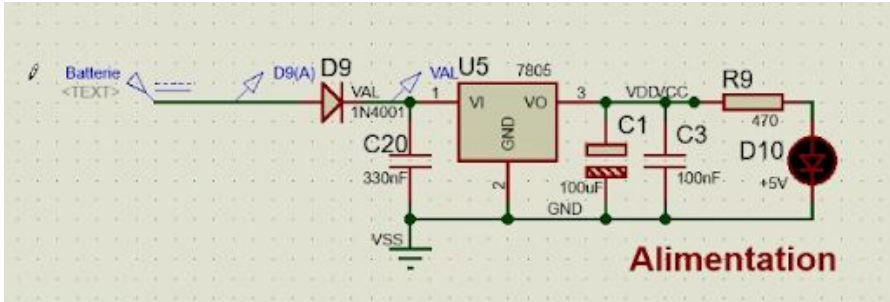
La diode verte D10 s'allume quand la carte est sous tension : cela confirme que les branchements sont corrects.

Vérification du régulateur U5 : en fixant Vbat à 12 puis 10V, le voltmètre indique toujours une tension stable à 5V ; le régulateur est fonctionnel.

En revanche, qd Vbat < 6.8V, la tension en sortie vaut 4.5 V : la tension n'est plus régulée, ce qui est cohérent avec la limite des 7 V vue en théorie.

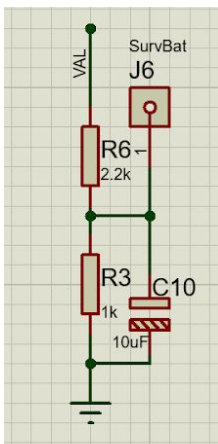
Le bloc est fonctionnel.

### Simulation Proteus :



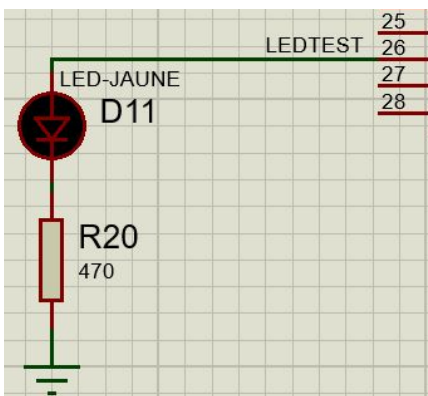
## Bloc surveillance alimentation

**Rôle du bloc :** Permettre au microcontrôleur U1 de connaître l'état de la batterie en lui fournissant une image de la tension de charge de la batterie.



On obtient la charge de la batterie en fonction de sa tension. Cette tension étant trop grande pour le microcontrôleur, on l'adapte alors avec un pont diviseur de tension (R6, R3) qui garantit une tension entre 0 et 5 volts ; à 12V,  $V_{R3}=3,75V$ . Le bloc alimentation vérifie alors que la tension est suffisamment élevée pour faire fonctionner le robot.

Vérification de bon fonctionnement de la réalisation :

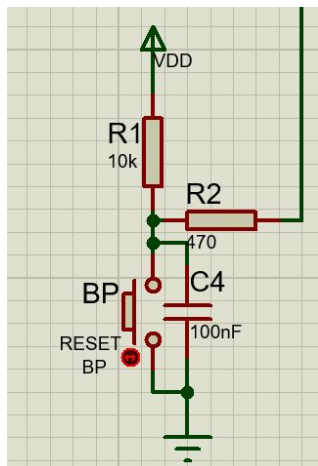


La diode jaune D11 s'allume pour une tension VBAT inférieure à 10V, tension de seuil basse pour que le régulateur renvoie 5V. Cela indique que les capteurs sont fonctionnels.

On relève au multimètre que  $V_{R3}=3.6\text{ V}$  (environ= à 1/3 de 12V ): cohérent avec la théorie, c'est à dire compris entre 0 et 5V.  
Le bloc est fonctionnel.

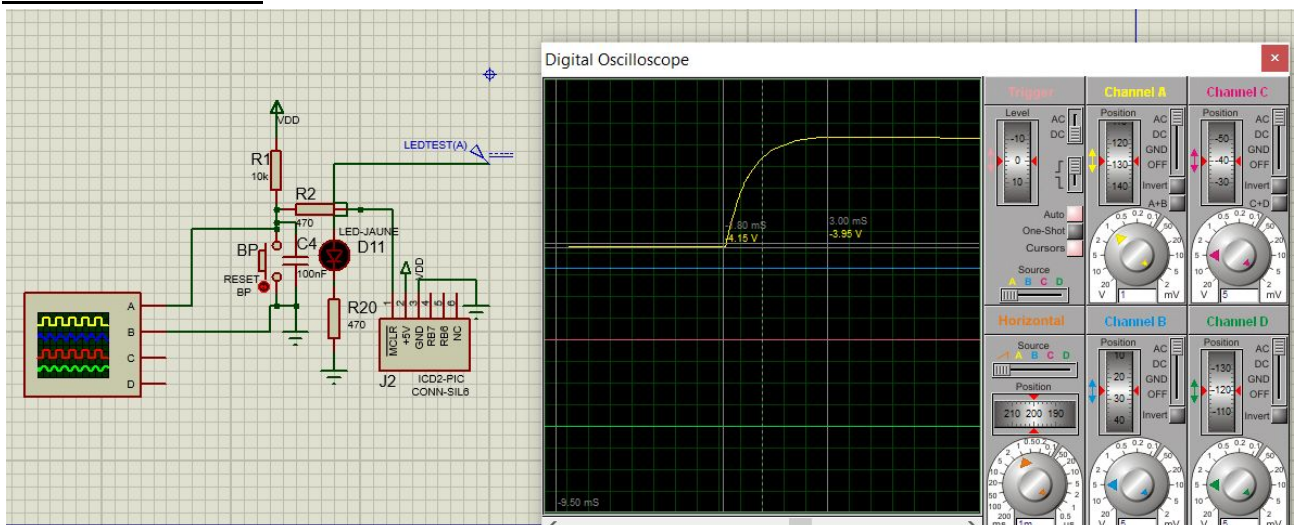
## Bouton reset

Rôle : le bouton presseur BP sert à reset le microcontrôleur sans le mettre hors tension.



Pour éviter de reset la carte par accident il y a une sécurité, constituée par la capacité C4 qui doit se vider complètement avant de lancer un redémarrage de la carte. En effet, appuyer sur le bouton décharge la capacité.

## Simulation Proteus :

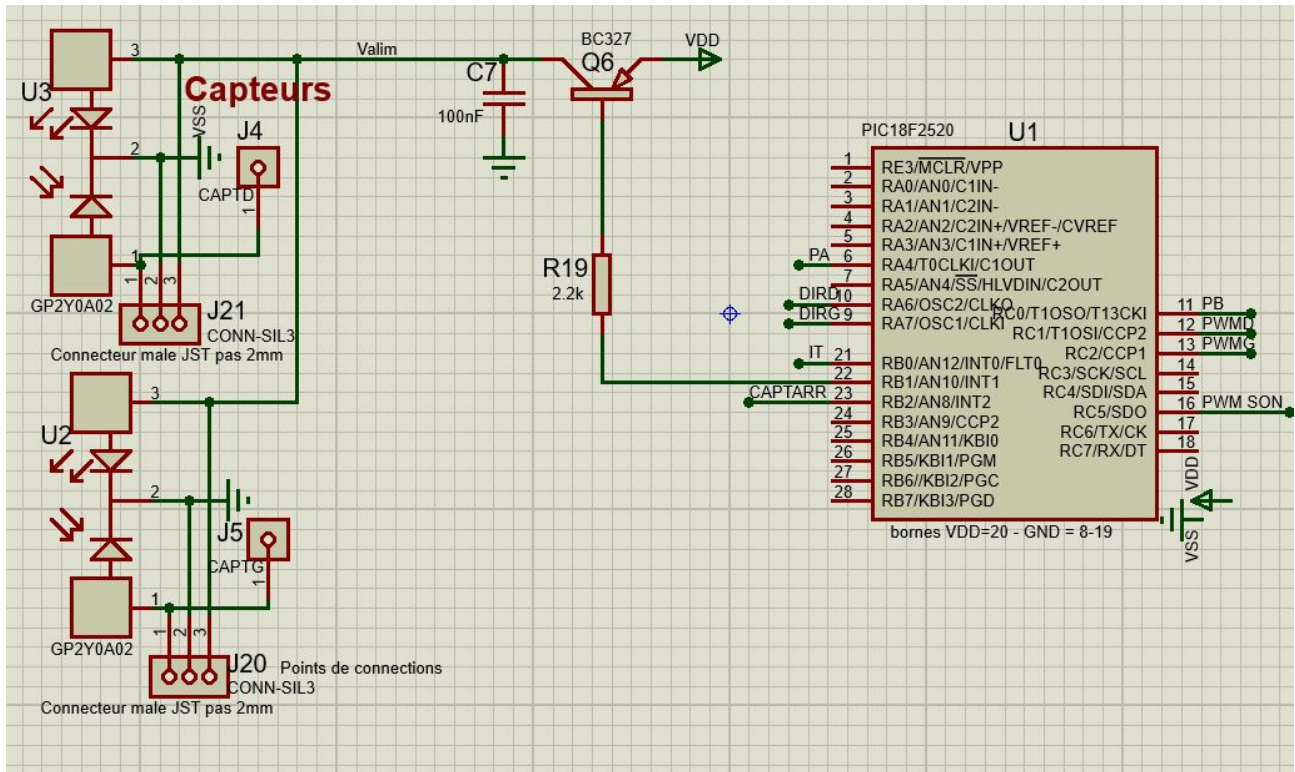


L'oscilloscope affiche la tension aux bornes du bouton poussoir BP. La courbe jaune montre que la durée de montée en tension après avoir enclenché BP vaut environ 3 ms. Pour amorcer un redémarrage sans débrancher la carte, l'utilisateur doit donc appuyer plus de 3 ms sur BP.



### III) Capteurs IR

Rôle du bloc : Il permet d'une part l'allumage et l'extinction des capteurs, et d'autre part de recueillir la mesure de distance entre les capteurs et l'objet auquel ils font face.



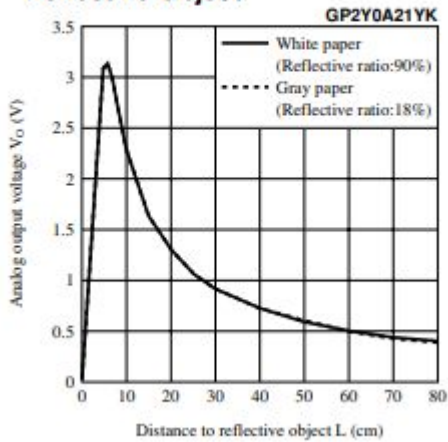
Les deux capteurs infrarouges sont connectés en J20 et J21 (à gauche de la carte) et alimentés par  $VDD = 5V$  avec une tolérance de  $0.5V$ .

Les capteurs sont alimentés via le transistor Q6 qui sert d'interrupteur: si U1 renvoie un 0 (on), les capteurs sont allumés et si U1 renvoie un 1 (off), ils sont éteints.

Les données des capteurs sont transmises en signaux analogiques (la tension analogique est croissante à mesure qu'un objet se rapproche) que le microcontrôleur mesure avec une précision de  $20\text{ mV}$ . Les capteurs IR utilisés sont destinés à mesurer des distances de  $5$  à  $80\text{ cm}$ , mais en pratique ils ont une portée maximale de  $24\text{ cm}$  (distance de jugement) et une portée minimale de  $5\text{ cm}$ . En effet, la tension analogique renvoyée par le capteur chute lorsque l'objet est à moins de  $5\text{ cm}$  du capteur. Ainsi le capteur ne peut déterminer si l'objet est proche ou loin pour une distance inférieure à  $5\text{ cm}$ .

Les informations analogiques des capteurs passent par les voies CAPTD et CAPTG qui permettent à U1 de traiter les signaux analogiques.

**Fig.5 Analog Output Voltage vs. Distance to Reflective Object**



Photographie du capteur IR SHARP GP2Y0A

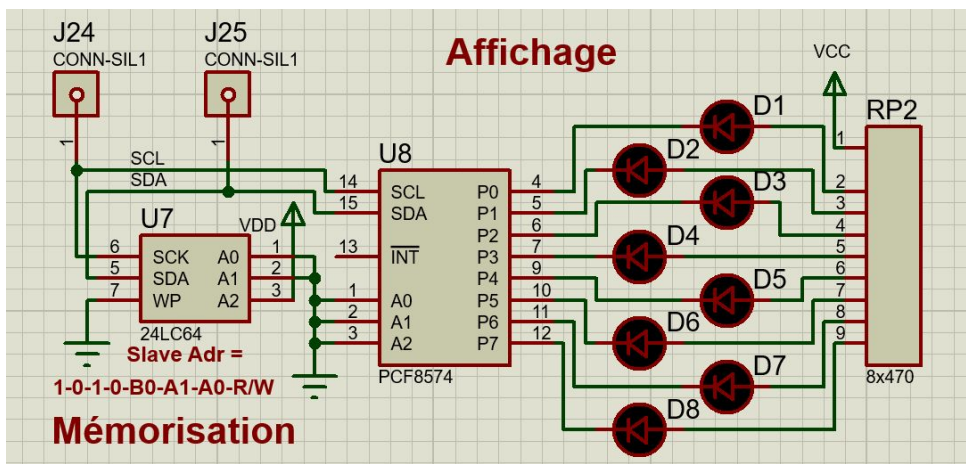
#### IV) Bloc affichage LED

Rôle du bloc : Afficher les informations de test du robot pour vérifier son bon fonctionnement. Il sert d'interface homme/machine.

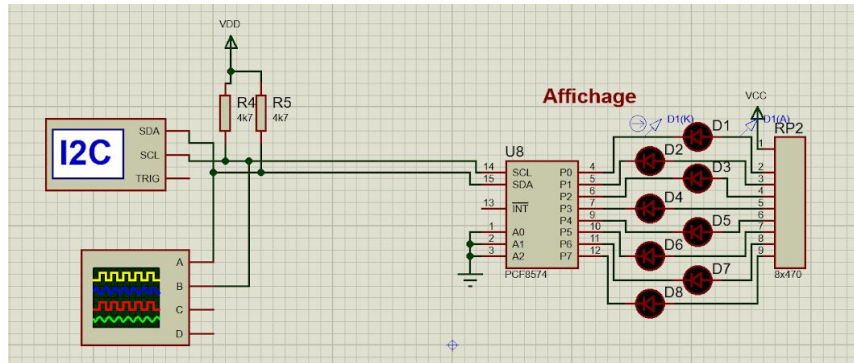
Quand la carte est en fonctionnement, l'utilisateur ne peut savoir ce qui s'y déroule sans un dispositif d'affichage, qui participe à l'interface utilisateur/carte.

Ce dispositif est composé du microcontrôleur U8, esclave du microcontrôleur U1, et de huit LED.

Le bus I<sup>2</sup>C permet au microcontrôleur de contrôler les huit diodes : en mettant 0V, U8 allume une LED (à cause de la différence de potentiel car V<sub>CC</sub>=5V); U8 éteint une LED en lui imposant une tension de 5V. Ce système de LED permet d'écrire en binaire, avec D1 le bit de poids faible et D8 celui de poids fort. Le système de mémorisation est composé de J24, J25 et U7; il sauvegarde l'état des LED, au cas où le programme serait interrompu.



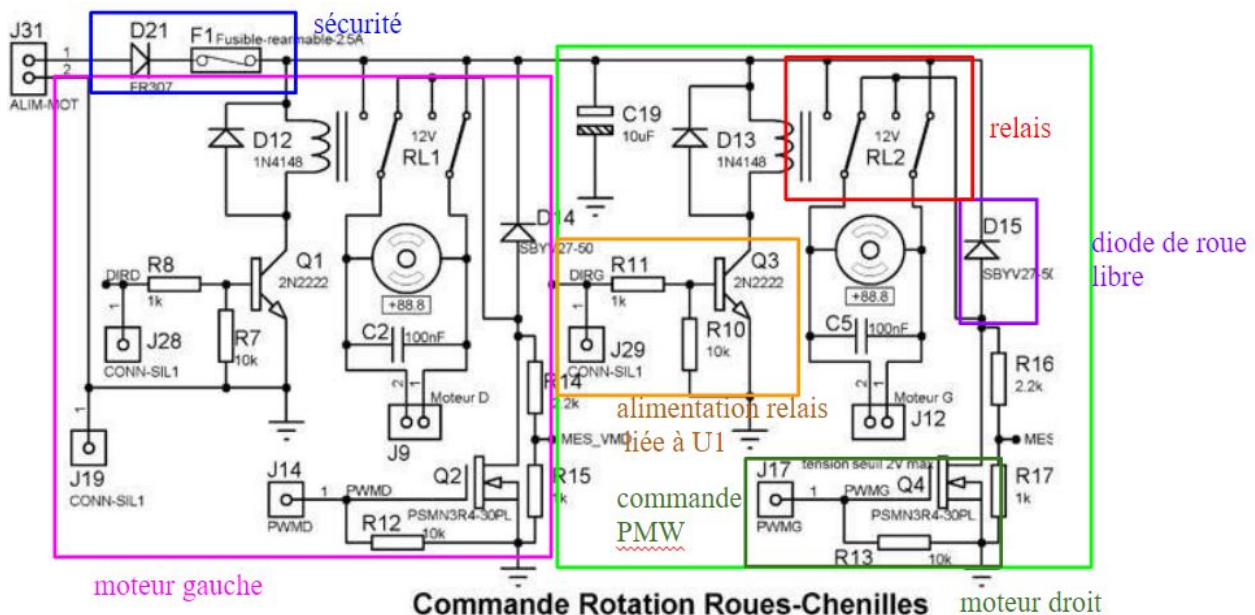
## Simulation Proteus



On manœuvre l’affichage LED de la même façon que le fera le logiciel de la carte robot.

## V) Moteurs/ Pilotage Rotation Roues-Chenilles

Rôle du bloc : Orienter et faire avancer le robot de la manière qu’a prévue le microcontrôleur.



Ce bloc est composé de deux sous-blocs identiques et de même fonction : commander le sens de rotation et la vitesse de chaque moteur.

Un sous-bloc contrôle le moteur droit, et l’autre contrôle le moteur gauche. Les moteurs ne sont pas alimentés par la carte, mais par un branchement à la batterie en J31 de façon à disposer de plus de puissance. Les blocs étant identiques, on ne décrira qu’un seul des blocs.

### **Protection du dispositif**

Les diodes D12 et D14 sont des diodes de roue libre : lorsque le transistor s’ouvre, ces diodes permettent de maintenir un circuit fermé pour ne pas abîmer les composants; elles protègent le circuit contre une surtension causé par une bobine lorsque le courant est coupé soudainement.

Le condensateur C2 supprime les hautes fréquences.

La diode D21 contrôle la polarité, ce qui protège le moteur.

Le fusible F1 protège les composants fragiles (comme les transistors) contre un courant trop élevé.

### **Sens de rotation des moteurs**

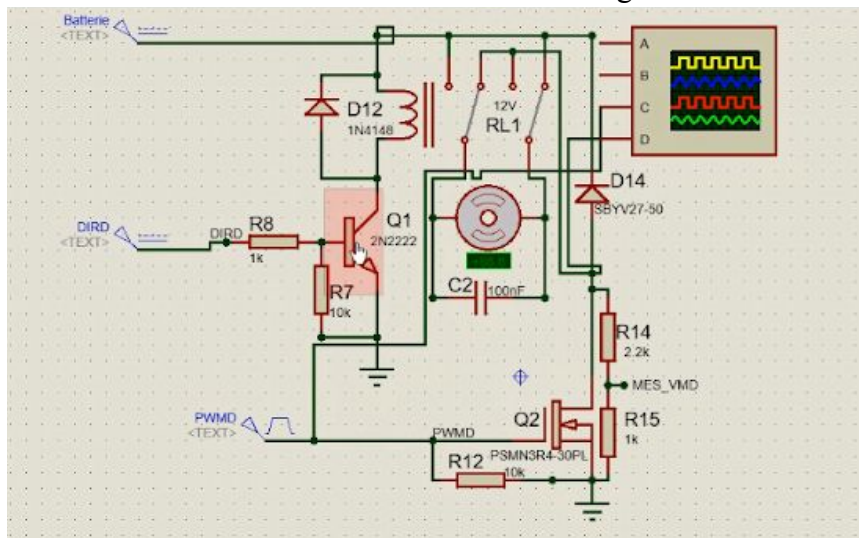


Le relais RL1 sert à inverser le signe de la tension, ce qui inverse le sens de rotation du moteur. RL1 est actionné par un transistor via une commande. En effet, le transistor TTL Q1 est ouvert en l'absence de courant et fermé sinon. Ce transistor a pour but de générer un champ électromagnétique qui fait tourner l'hélice du relais RL1. La commande envoyée par le microcontrôleur via la sortie DIRD est reçue en J28 pour le moteur gauche. Le sens de rotation est par défaut, trigonométrique, lorsque Q1 est passant. Enfin, basculer la tension DIRD de 0 à 5V permet de changer le sens de rotation.

### Vitesse de rotation

La voie PWMD s'occupe de la vitesse du moteur en J14. La tension au niveau de la voie PWMD est gérée par le microcontrôleur. En effet, il modifie le rapport cyclique du signal pour augmenter ou diminuer la tension moyenne. Cette tension moyenne est la tension de grille de Q2, donc la modifier vient à modifier la tension drain-source et donc la vitesse de rotation. En synthèse, la vitesse de rotation du moteur est proportionnelle à l'épaisseur du créneau WMD

Simulation sur Proteus du moteur gauche

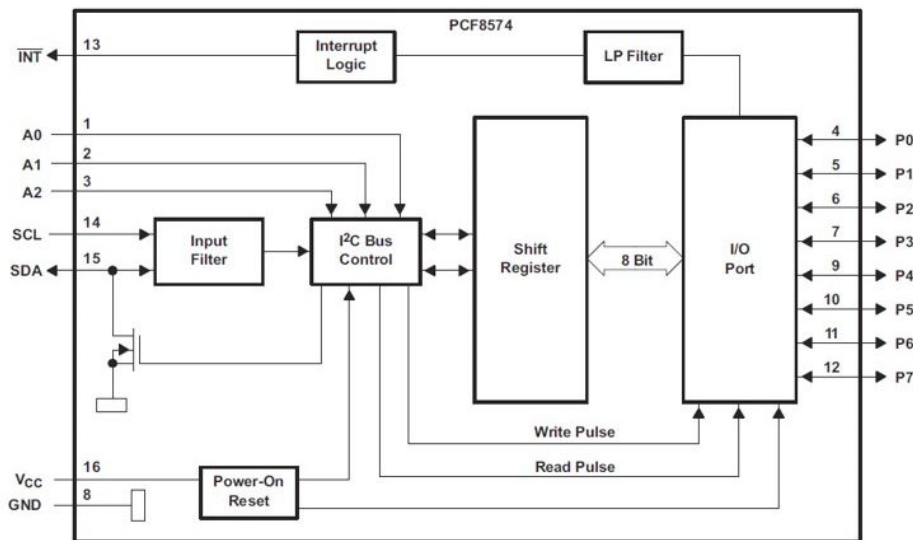


## VI) Affichage bus I<sup>2</sup>C (Inter-integrated circuit)



**Rôle du bloc :** Communiquer avec les périphériques (les microcontrôleurs U8 et U1), afficher les données envoyées et reçues.

*Schéma simplifié du PCF8574*



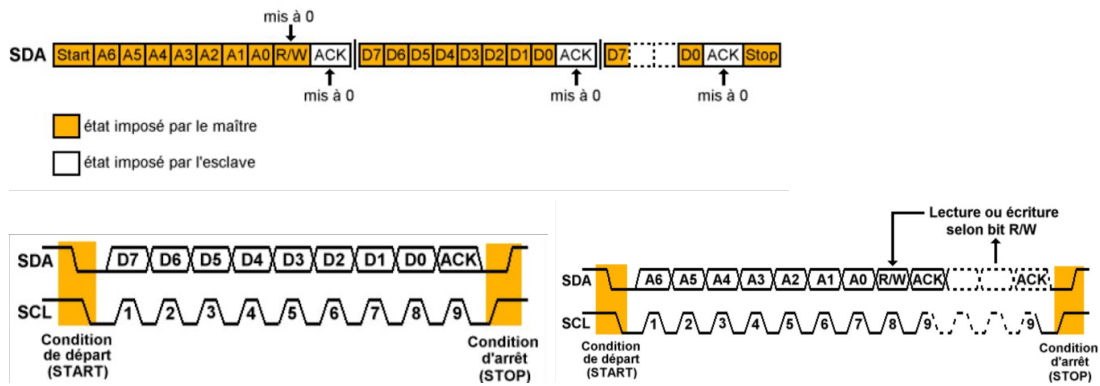
Le bus de pilotage I<sup>2</sup>C est constitué par le composant PCF8574. Le protocole I<sup>2</sup>C permet la transmission de données entre le microcontrôleur et les différents périphériques. A2, A1 et A0 portent respectivement l'état de la masse, de SDA et de la clock (SCL).

C'est le maître (le microcontrôleur) qui envoie le START et le STOP d'une communication ainsi que la condition R ou W. L'esclave (le périphérique) quant à lui reçoit seulement.

S est le signal start, les bits A6 à A0 composent l'adresse du périphérique et les bits D7 à D0 composent la donnée.

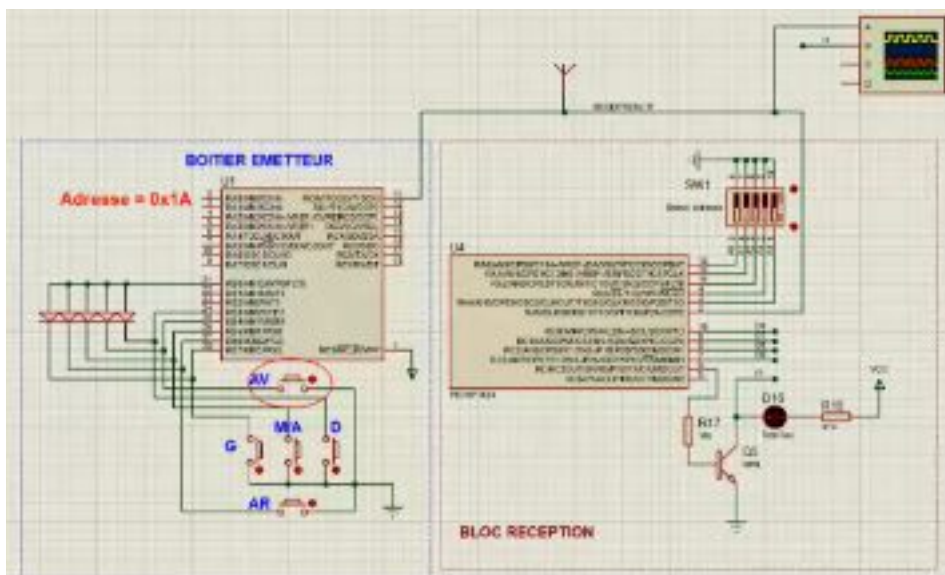
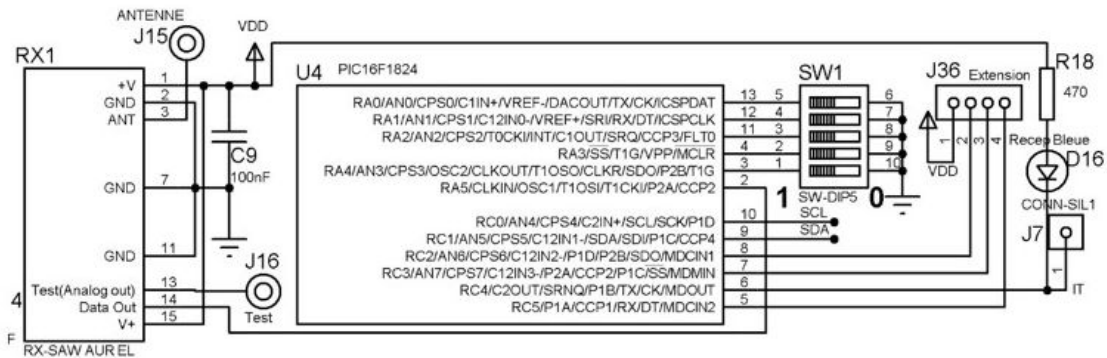
Un bit pour savoir si on veut lire ou écrire (R/W) = 1 pour Read et = 0 pour Write. Le bit ACK pour acknowledged si ACK est lu à 1 (= NACK), il y a une erreur, l'esclave ne se reconnaît pas, le maître renvoie une erreur.

L'esclave reconnaît un message par un front descendant sur SDA puis un sur SCL à la suite, cela permet de vérifier que le bus est libre. La fin d'un message est l'opposé, un front montant sur SCL puis un sur SDA. Les communications sont envoyées en double pour que l'esclave les comprenne bien.

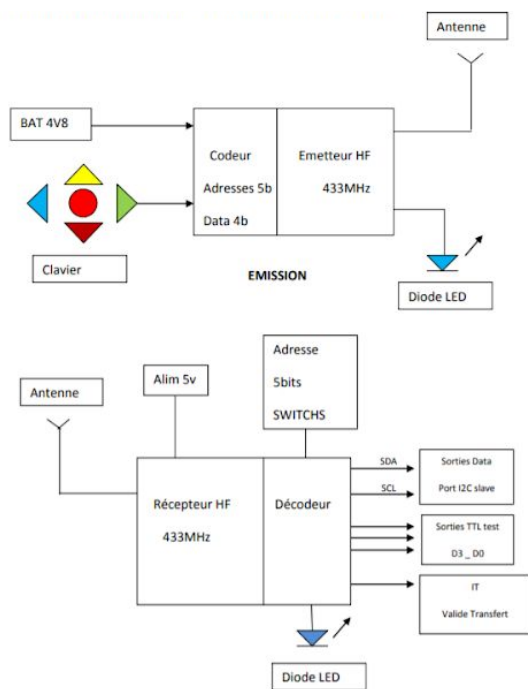


## VII) Réception télécommande

### Télécommande



Rôle du bloc : Donner à l'utilisateur le contrôle du robot sur 5 instructions basiques.

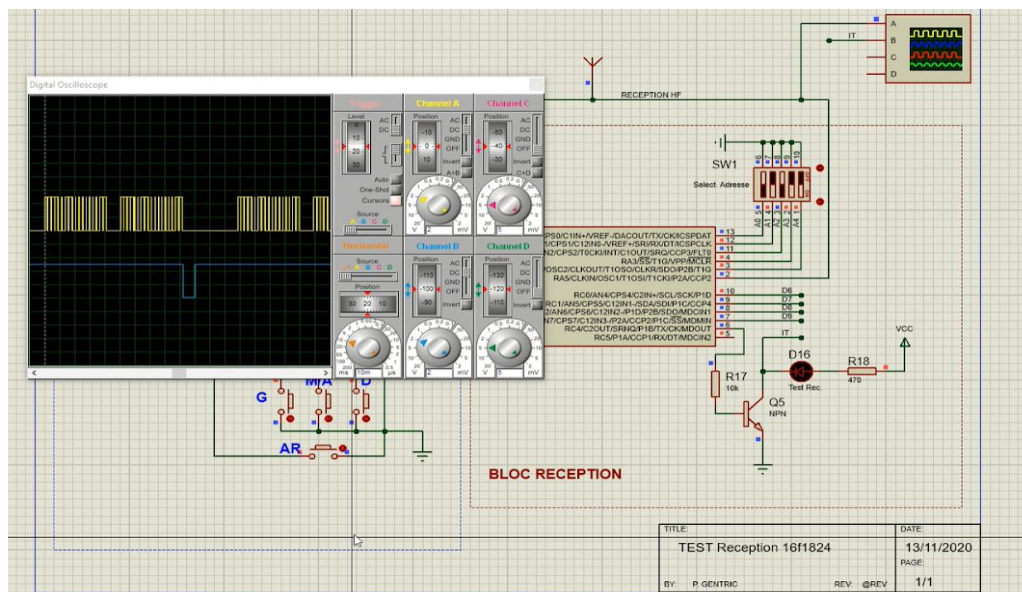


Les 5 touches (en couleur sur le schéma) permettent de déplacer le robot selon les 4 directions et d'allumer ou d'éteindre le robot à distance. Le rôle de l'antenne est assuré par un connecteur, qui capte les informations envoyées par la télécommande (c'est une antenne quart d'onde, elle mesure un quart de la longueur d'onde du signal envoyé).

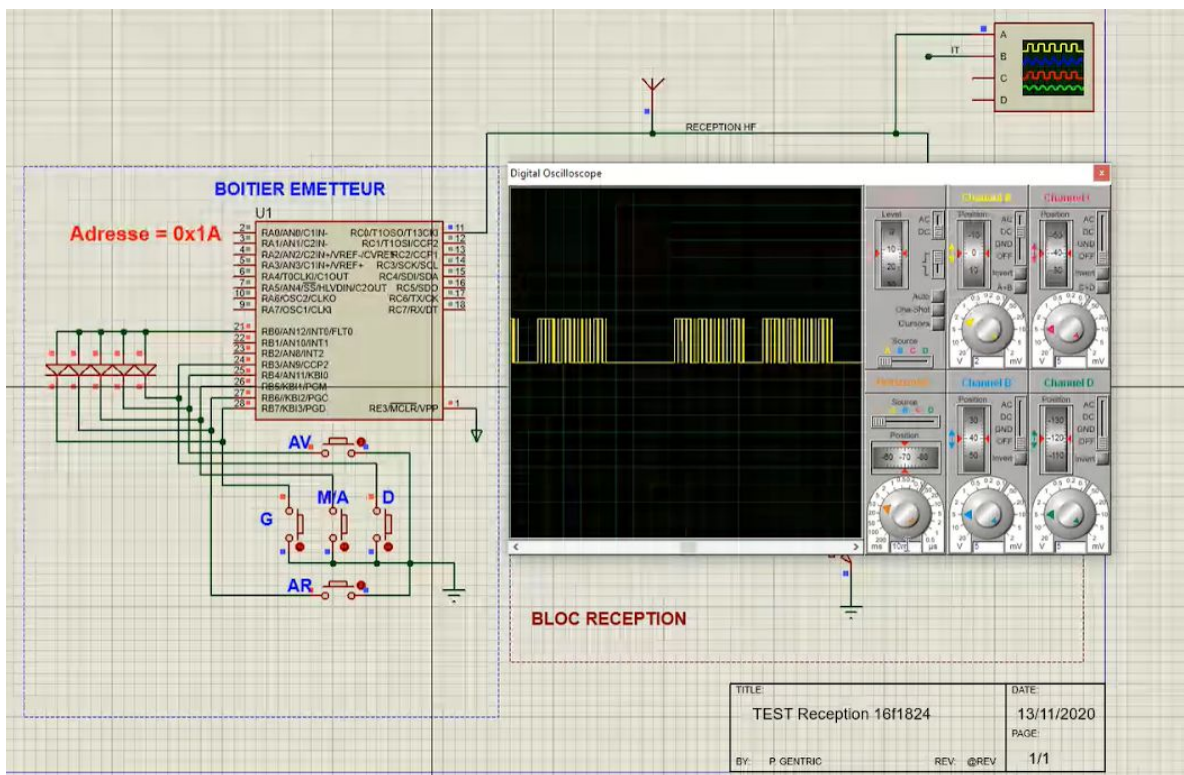
Lorsqu'un signal est perçu par l'antenne sur la carte, RX1 convertit ce signal en codage OOK (On-Off Keying) qui est une façon de communiquer en binaire. Il est ensuite transmis à U4 (en format TTL 5V) qui se chargera de le convertir en signal analogique. Enfin, en vérifiant grâce au switch SW1 que la télécommande est la bonne, U4 transmettra les informations à l'I²C via SDA et SCL. L'adresse du récepteur est programmée par les cinq micro-switches de DSW1. La diode D16 témoigne de la réception d'un bon signal.

Simulation du bloc télécommande sur Proteus (après avoir actionné la touche T1):

Les trames de chaque touche étant uniques, cela permet au microcontrôleur de déterminer la touche pressée par l'utilisateur.



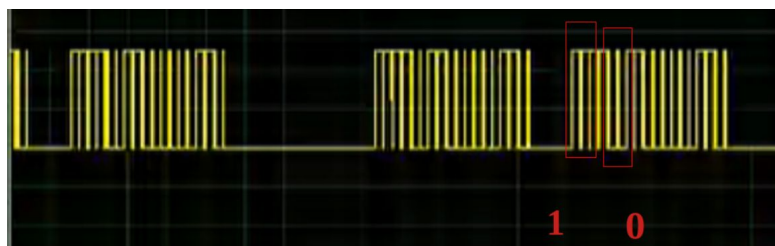




L'oscilloscope permet de visualiser le signal reçu de la télécommande.

On mesure à l'aide de l'oscilloscope et en appuyant sur la touche T1 de la télécommande plusieurs paramètres, dont la durée d'une trame en entrée du boîtier récepteur, ainsi que le codage à effectuer sur SW1 en sortie du récepteur.

On peut déchiffrer le code envoyé par la télécommande grâce à la conversion de binaire en décimal sur l'oscilloscope branché au bloc Réception télécommande.



trame reçue

On modifie les interrupteurs de SW1 (01011). On relève avec l'option Cursors qu'une trame dure 92 ms. La diode D16 clignote rapidement au rythme de celui du boîtier de la télécommande quand on appuie sur le bouton AV.

Cela témoigne de la bonne réception du signal.

## VIII) Sonar et motorisation du sonar par le servomoteur

### Sonar



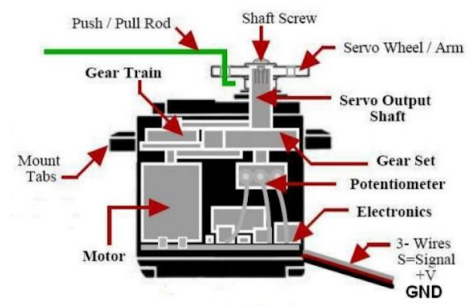
photographie du sonar

Le robot est équipé d'un sonar, qui communique avec le microcontrôleur via le bus I<sup>2</sup>C. Le sonar est équipé d'une carte qui effectue le traitement des données qu'il génère. Le sonar est un capteur directif (avec un angle de vue de 60°), dont le champ d'action s'étend de 10 cm à plus d'un mètre.

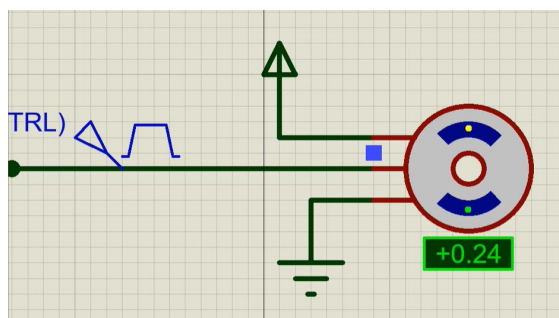
### Servomoteur

Rôle du bloc : Le servomoteur permet de commander l'orientation sur 185° du sonar, ce qui lui permet de visualiser tout l'environnement à sa portée.

photographie (à gauche)  
et schéma (à droite) du servomoteur

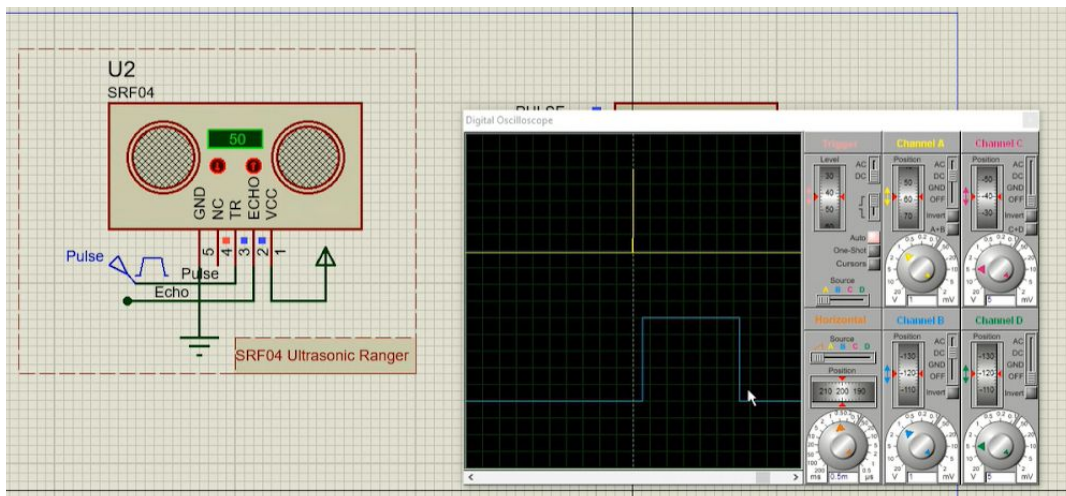


### Simulation Proteus du servomoteur seul



Le servomoteur est un organe mécanique commandé par le microcontrôleur au moyen d'une consigne d'angle. Il reçoit un signal PWM qui le fait tourner de 5 en 5°, ses angles extrêmes sont 95° et -90°.

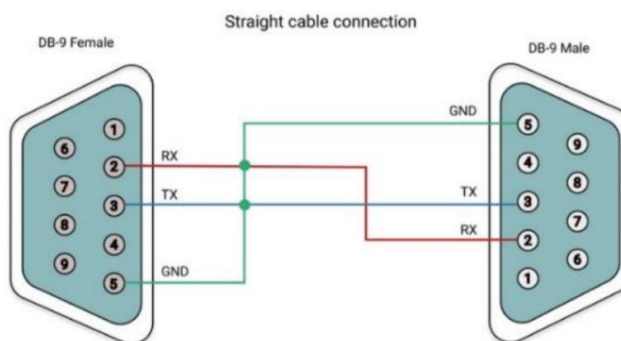
### Simulation sur Proteus:



On simule le fonctionnement du sonar pour un mur à une certaine distance, on récupère le temps d'un aller retour (c'est l'épaisseur du créneau) et on calcule ainsi la distance au mur.

## IX) Étude de la liaison RS232

Le standard RS232 est une norme qui standardise un bus de communication de type série sur au moins trois paramètres (mécanique, électrique, et protocole). Cette méthode de communication entre deux équipements est bidirectionnelle (échange en duplex), asynchrone (pas de signal d'horloge), série (les données sont sérialisées) et unilatérale (chaque récepteur est relié à un seul émetteur).



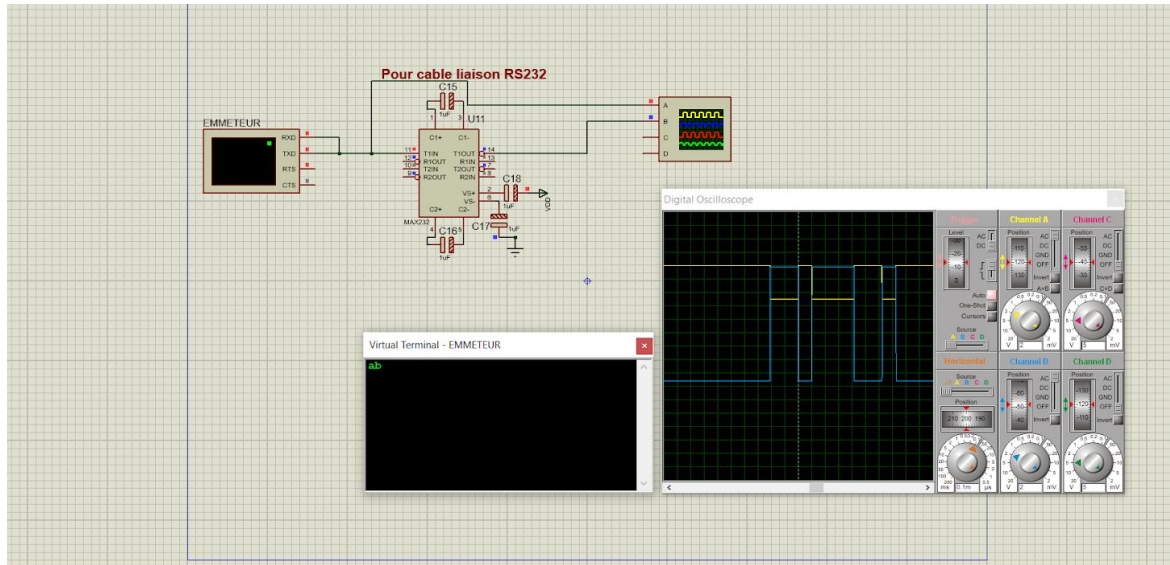
La transmission série nécessite un minimum de 2 fils comportant les trames de données en émission (Tx) et en réception (Rx).

En outre, les paramètres qui rentrent en jeu dans les protocoles de transmission RS232 sont caractérisés par la vitesse de transmission (appelée baudrate, ou nombre de bits par seconde), la parité, le bit de départ, le nombre de bit d'arrêt (1 ou 2 bits de stop), le bit de parité (optionnel) et les bits d'information qui composent les caractères (la longueur des mots). Ces derniers font de 5 à 8 bits par caractères et sont émis du plus faible poids, au plus élevé.

L'adaptation des données se fait à l'aide d'un circuit adaptateur de ligne MAX232, qui transforme les niveaux logiques issus du système numérique en niveaux logiques compatibles avec les normes

RS232 et vice versa. C'est ce circuit qu'on teste dans la simulation Proteus ci-dessous.

### Simulation Proteus: simulation de MAX232



Sur le fichier Proteus, on relie RxD et TxD pour que le terminal virtuel (Virtual Terminal) reçoive la commande venant du microcontrôleur.

On tape des caractères dans la fenêtre de l'émetteur ici "ab"), et on observe en bleu le signal reçu; ce signal est en opposition de phase au signal émis et possède une amplitude supérieure.

Une précaution à prendre par l'utilisateur est de s'assurer que les réglages de l'émetteur et du récepteur sont les mêmes; voici les problèmes rencontrés en cas de réglages différents :

en premier lieu, lorsque l'émetteur et le récepteur n'ont pas les mêmes réglages de parité (par exemple émetteur pair sur 2 bits et récepteur impair 1 bit), l'affichage est rempli de caractère sans aucun sens, l'émetteur et le récepteur ne peuvent pas communiquer de façon fonctionnelle.

En second lieu, lorsque l'émetteur envoie plus de bits que le récepteur n'est censé en capter (par exemple l'émetteur sur 8 bits et le récepteur sur 7 bits), cela a pour conséquence de rajouter un caractère à l'affichage, (caractère qui correspond au huitième bit pour l'exemple). En sens inverse l'émetteur reçoit moins de bit que prévu il affiche donc n'importe quoi.

En troisième lieu, lorsque l'émetteur est réglé à 2 bits de stop et le récepteur à 1 bit, écrire depuis l'émetteur fonctionne car le récepteur lit un bit de stop et s'arrête. En sens inverse l'émetteur ne lit qu'un bit de stop et attend le deuxième.

Enfin, lorsque le baudrate n'est pas réglé sur la même valeur pour l'émetteur et pour le récepteur, la durée de la communication n'est pas la même pour les deux. Les résultats sont donc absurdes puisque l'un arrête d'émettre alors que l'autre écoute encore ou à l'inverse il arrête d'écouter alors qu'il émettait.



## X) Conclusion

Le projet robot nous a donné une vision globale de la réalisation d'un projet électronique. En effet, l'étude des différents blocs et l'utilisation du logiciel Proteus nous a permis de mettre en application nos connaissances théoriques. Mais aussi de comprendre certaines subtilités, comme la diode de roue libre pour n'en citer qu'une.

Bien que nous n'ayons pas pu finir la carte en raison des circonstances, le projet robot nous a donc permis de comprendre de manière pratique des notions du cours d'électronique. Mais aussi d'affiner nos compétences de travail en binôme.

Cette expérience fut une excellente introduction à cet univers et nous attendons avec impatience la possibilité d'appliquer ce que nous avons appris dans un cas plus pratique que des simulations sur Proteus.